素数判定の決定的多項式時間アルゴリズム (代数的整数論とその周辺)

AUTHOR(S):

木田, 雅成

CITATION:

木田, 雅成. 素数判定の決定的多項式時間アルゴリズム (代数的整数論とその周辺). 数理解析研究所講究録 2003, 1324: 22-32

ISSUE DATE:

2003-05

URL:

http://hdl.handle.net/2433/43143

数理解析研究所講究録 1324 巻 2003 年 22-32

素数判定の決定的多項式時間アルゴリズム

木田雅成 (Masanari Kida)
電気通信大学
(University of Elecctro-Communications)

本稿の目的は 2002 年 8 月 6 日付の Agrawal, Kayal, Saxena の論文 [1] で発表された決定的多項式時間の素数判定アルゴリズムについて解説することである. 始めに第一節で歴史を簡単に振り返りながら, 基本的な定義を思い出す. 第二節では AKS のアルゴリズムの根拠となる定理を証明つきで解説する. 第三節で AKS のアルゴリズムを紹介する. 最後の節では計算量と実装例について述べる.

1 基本的な定義と歴史

アルゴリズムの複雑さ (計算量) は何度基本的な演算を行なうかを入力の長さ (あるいは桁数) $\log N$ の関数としてはかる. 計算量が $\log N$ の多項式になるとき多項式時間アルゴリズムという. [11] の第二章に計算量の理論の簡単な解説がある.

以下ではこれまで知られていた結果について概観する. AKS 以前に知られているアルゴリズムについては [5] が詳しい.

1.1 Miller-Rabin test (1980)

命題 1 ([13],[14]). N を正の整数とする. $N-1=2^s\cdot d$, $(d,2)=1$ と書く. 次の条件をみたす整数 a があれば N は合成数である.

$$(a,N)=1,\quad a^d\not\equiv 1 \pmod N,\quad a^{2^r d}\not\equiv -1 \pmod N\ \forall r=0,\dots,s-1. \tag{1.1}$$

これをアルゴリズムにすると, 次のようになる.

Input: 整数 $N>1$.

1: $N-1=2^s d$ と $(d,2)=1$ をみたす s,d を計算する.
2: ランダムに $a\in[2,n-2]$ を選ぶ.
3: $b:=a^d \pmod N$;
4: **if** $b\neq 1$ or $b\neq -1$ **then**
5: **return composite**;
6: **end if**
7: **for** $r=1$ to $s-1$ **do**

8: $b := b^2 \pmod{n}$;
9: **if** $b \neq -1$ **then**
10: **return** `composite`;
11: **end if**
12: **end for**
13: **return** `probably prime`;

このアルゴリズムが, N が合成数であるときに, `probably prime` を返す確率は a のランダムな選択に関して 1/4 以下であることが知られている.

このようにランダムな整数の選択を含み, その選択に依存して答えを返す多項式時間アルゴリズムで正しい確率が 1/2 以上のものを確率的多項式時間アルゴリズムという. Miller-Rabin のアルゴリズムは「合成数判定問題」の確率的多項式時間アルゴリズムである. また, このようなランダムな選択を含まない多項式時間アルゴリズムを決定的多項式時間アルゴリズムという. [1] の論文の題名 'PRIMES is in P' は素数判定問題には決定的多項式時間アルゴリズムが存在することを示す.

さらに Dirichlet の L 関数に関する Riemann 予想を仮定すると, N が合成数の時 (1.1) をみたす a が $2(\log N)^2$ 以下でとれることが知られているので, 上のアルゴリズムの 2 行目を

$$\textbf{for } a = 2 \textbf{ to } 2(\log N)^2 \textbf{ do}$$

のループに変えることによって, 「素数判定問題」の決定的多項式時間アルゴリズムになる.

1.2 その他の素数判定法

暗号理論などに使われる素数の生成には Miller-Rabin test を何度か繰り返して使うことで十分なことが多い. また KASH や PARI, MAGMA といった数論ソフトウェアで使われているのもこのアルゴリズムである. $a = 2, 3, \ldots, 17$ の 7 個の数に対して Miller-Rabin test を適用すれば $n < 341550071728321$ の数に対しては正しい答が得られるという [9] の結果も MAGMA では併用しているようである.

他にも以下のような素数判定法がある. 特に前二者の判定法を使うと, 十進数千桁の素数判定が実行可能である.

Adleman-Pomerance-Rumely (1983) ヤコビ和, ガウス和の性質を使った決定的アルゴリズムである. ただし計算量は多項式時間ではなくて $O((\log N)^{C \log\log\log N})$ である.

Goldwasser-Kilian (1986) 楕円曲線を使った素数判定法. 計算量に関しては heuristic な結果しか知られていない.

Adleman-Huang (1992) 種数 2 の超楕円曲線を使った素数判定法. 素数判定問題の確率的多項式時間アルゴリズムである.

2 AKS の定理

AKS のアルゴリズムは以下の定理に基づいている.

定理 2 (Agrawal and Kayal and Saxena [1]). N を 1 より大きい整数とする. 次の条件をみたす素数 q, r と, 正の整数 s があれば N は素数冪である.

(1) $q \mid (r-1)$
(2) $N^{\frac{r-1}{q}} \not\equiv 0 \text{ nor } 1 \pmod{r}$
(3) N は s よりも小さい約数を持たない.
(4) $\binom{q+s-1}{s} \geq N^{2\lfloor\sqrt{r}\rfloor}$
(5) $(x-a)^N \equiv x^N - a \pmod{(N,\, x^r-1)}$ が $a = 1, 2, \ldots, s$ についてなりたつ.

証明. 以下の証明は Bernstein の [3] に基づいている. (1) より $\dfrac{r-1}{q}$ は整数で, (2) より N の素因数 p で

$$p^{\frac{r-1}{q}} \not\equiv 0 \text{ nor } 1 \pmod{r} \tag{2.1}$$

をみたすものがある. ここで $p^i N^j$, $(0 \leq i, j \leq \lfloor\sqrt{r}\rfloor)$ を考えると, r 個以上あるので,

$$\exists (i,j),(k,\ell) \begin{cases} (i,j) \neq (k,\ell) \\ p^i N^j \equiv p^k N^\ell \pmod{r}. \end{cases}$$

$u = p^i N^j$, $v = p^k N^\ell$ とおく. このとき $u = v$ がいえれば, $(i,j) \neq (k,\ell)$ だから N が p の冪になって証明終り.

そのために, 次の (i) と (ii) をみたす巡回群 $G = \langle g \rangle$ の存在を示す.

(i) $\#G > N^{2\lfloor\sqrt{r}\rfloor}$,
(ii) $g^u = g^v$.

この二つの条件が成り立てば $u = v$ となることをみる. 実際, (ii) から $g^{u-v} = 1$ であって,

$$p^i N^j \leq N^{i+j} \leq N^{2\lfloor\sqrt{r}\rfloor}$$

がこの形の任意の元についてなりたつので,

$$|u - v| \leq N^{2\lfloor\sqrt{r}\rfloor}.$$

よって, (i) より $u = v$.

この 2 条件をみたす群 G を次のように構成する. $h(x) \in \mathbb{F}_p[x]$ を r 番目の円分多項式 $\dfrac{x^r-1}{x-1}$ の mod p での既約因子として,

$$G = \langle x - a \mid a = 1, 2, \ldots, s \rangle \subset \left(\mathbb{F}_p[x]/(h(x))\right)^\times$$

とする. G は巡回群であって, 上の条件 (i) と (ii) をみたす. 以下の証明では右辺の体を $\mathbb{K}$ と書く.

まず (i) を示す. 円分体の理論から $\deg h = \mathrm{ord}(p \bmod r)$ であるが, (2.1) より, $\mathrm{ord}(p \bmod r)$ は q の倍数. 特に $\deg h \geq q \geq 2$. よって $x - a \neq 0$ in $\mathbb{K}$. また $a, a' \in \{1, 2, \ldots, s\}$ で $a \neq a'$ とする. このとき $x - a = x - a'$ が $\mathbb{F}_p[x]$ でなりたつならば, $p|(a - a')$ となり, 右辺の絶対値は s より小さいので (3) に反する. よって $x - a \neq x - a'$. さて, ここで,

$$\prod_{a=1}^{s}(x-a)^{e_a}, \quad \left(\sum_{a=1}^{s} e_a \leq q-1\right) \tag{2.2}$$

を $\mathbb{F}_p[x]$ で考える. これらの元はすべて $\mathbb{K}$ の中で異なる元であることを示そう. 実際, このような形をした二つの元が $\bmod h(x)$ で等しいとすると, $\deg h \geq q$ より, それらは $\mathbb{F}_p[x]$ で等しい. $x - a$ たちはすべて異なるので, 結局それらの冪も等しい. したがって, (2.2) はすべて異なる G の元. それらは $\binom{q+s-1}{s}$ 個あるので, (4) より, (i) がみたされる.

次に (ii) を証明する. そのために自然な準同型

$$A = (\mathbb{Z}/N\mathbb{Z})[x]/(x^r - 1) \longrightarrow B = \mathbb{F}_p[x]/(x^r - 1) \longrightarrow \mathbb{K} = \mathbb{F}_p[x]/(h(x)).$$

において A, B で関係を作って, $\mathbb{K}$ におとす. (5) で x を x^{N^j} に置き換えると

$$(x^{N^j} - a)^N \equiv x^{N^{j+1}} - a \pmod{x^{N^j r} - 1}.$$

よって $(x^{N^j} - a)^N = x^{N^{j+1}} - a$ が A でなりたつ. この式を使うと帰納法で

$$(x-a)^{N^j} = x^{N^j} - a \text{ in } A.$$

が示せる. 両辺を p^j 乗して, $\bmod p$ で考えると,

$$(x-a)^{N^j p^j} = x^{N^j p^j} - a \text{ in } B.$$

さて, 先にとった u, v は $u \equiv v \pmod r$ をみたしていて, A では $x^r = 1$ だから $x^u = x^v$ in A. また B では

$$(x-a)^u = x^u - a = x^v - a = (x-a)^v.$$

よってこれが $\mathbb{K}$ で成立する. G の元は $x - a$ たちの積だから, すべての $g \in G$ について $g^u = g^v$ が成立する. □

3 AKS algorithm

定理2に基づいてアルゴリズムを書くと次のようになる.

Input: 整数 $N > 1$

1: **if** $N = m^k$ with $k > 1$ **then**
2: **return composite;**
3: **end if**
4: **for** $r = 2$ to $N - 1$ **do**
5: **if** r is a prime **then**
6: **if** $r|N$ **then**
7: **return composite;**
8: **end if**
9: $q :=$ ($r-1$ の最大素因子);
10: **if** $N^{\frac{r-1}{q}} \not\equiv 1 \pmod{r}$ **then**
11: **for** $s = 1$ to $q - 1$ **do**
12: **if** $\binom{q+s-1}{s} \geq N^{2\lfloor\sqrt{r}\rfloor}$ **then**
13: **break;**
14: **end if**
15: **end for**
16: **end if**
17: **end if**
18: **end for**
19: **for** $a = 1$ to s **do**
20: **if** $(x-a)^N \neq x^N - a$ in $(\mathbb{Z}/N\mathbb{Z})[x]/(x^r - 1)$ **then**
21: **return composite;**
22: **end if**
23: **end for**
24: **return prime;**

このアルゴリズムは大きく三つの部分にわかれる. 第一の部分は1行目から3行目までで, N が冪になっていないかを調べる部分である. 第二の部分は4行目から18行目のループで (5) 以外の条件をみたす r, s の選択を行なう. これをみたす r, s の存在は次節で示すが, それがなくてもこのアルゴリズムは有限回で終了する. 第8行が終わった時点で N は r より小さい素因子を持たないから, $s \leq q \leq r$ より (3) のチェックは明示的にしなくてよい. 第三の部分は19行目から23行目のループでここで (5) のチェックをする. N が素数であれば, Fermat の小定理によりこのループは必然的に通り抜けることに注意する.

4 多項式時間であることの証明

整数や多項式の基本的な演算が $\log N$ の多項式のオーダーで実行可能であることを認めると，このアルゴリズムが多項式時間でになるには，r が $\log N$ の多項式の大きさでとれればよい．実際，そうならば4行めから18行めのループも，19行目以下のループも $s \le q-1 \le r-1$ であるから，$\log N$ の多項式の回数の繰り返しで終了する．以下ではこのことを証明する．そのために次の二つの補題を使う．

補題 3 (Goldfeld [7]. [2] もみよ). 正の定数 A と $\delta > 1/2$ があって，十分大きな x に対して

$$\#\{p \mid p \text{ は } x \text{ 以下の素数で } p-1 \text{ の最大素因子は } x^{\delta} \text{ より大きい}\} \ge A\frac{x}{\log x}.$$

補題 4. $\pi(x)$ を x 以下の素数の個数とする．正の定数 B があって，十分大きな x に対して

$$\pi(x) \le B\frac{x}{\log x}.$$

補題4をみたす B の存在は素数定理が証明される以前から知られていた．補題3の A と 補題4の B は以下で見るように計算量には本質的には影響を及ぼさないが，補題3の δ の値は本質的な影響を与える．これについては後でもう一度コメントする．

さて $s = [2[\sqrt{r}]\log_2 N] + 1$ とおき，$q \ge 2s$ を仮定すると，

$$\binom{q+s-1}{s} > \left(\frac{q}{s}\right)^s \ge 2^s \ge N^{2[\sqrt{r}]}$$

となり (4) が成立．また $q \ge 4\sqrt{r}\log_2 N$ なら上の s について，(4) はみたされる．よって多項式時間であることは次の定理から導かれる．

定理 5. 補題 4 の δ に対して，k を

$$k \ge \frac{2}{2\delta - 1} \tag{4.1}$$

をみたす整数とする．正の定数 c_1, c_2 をうまくとると，

$$\exists \text{素数 } r \in [c_1(\log N)^k, c_2(\log N)^k] : \begin{cases} (P1)\ r-1 \text{ は } q \ge 4\sqrt{r}\log_2 N \text{ をみたす素因子 } q \text{ をもつ} \\ (P2)\ N^{\frac{r-1}{q}} \not\equiv 1 \pmod{r} \end{cases}$$

証明. $\mathbb{P}(x)$ で x 以下の素数の集合を表し，$\mathbb{P}'(x)$ で x 以下の素数で補題3の最大素

因子の条件をみたすものの集合を表すことにすると,

$$\begin{aligned}
\#(\mathbb{P}'(c_2(\log N)^k) \cap [c_1(\log N)^k, c_2(\log N)^k]) &\geq \#\mathbb{P}'(c_2(\log N)^k) - \#\mathbb{P}'(c_1(\log N)^k) \\
&\geq \#\mathbb{P}'(c_2(\log N)^k) - \#\mathbb{P}(c_1(\log N)^k) \\
\text{(補題 3 と 4 より)} &\geq \frac{Ac_2(\log N)^k}{\log(c_2(\log N)^k)} - \frac{Bc_1(\log N)^k}{\log(c_1(\log N)^k)} \\
(c_1 \geq 1 \text{ とすると}) &\geq \frac{Ac_2(\log N)^k}{\log(c_2(\log N)^k)} - \frac{Bc_1(\log N)^k}{k\log(\log N)} \\
\text{(十分大きな } N \text{ に対して)} &\geq \frac{Ac_2(\log N)^k}{(k+1)\log\log N} - \frac{Bc_1(\log N)^k}{k\log\log N} \\
&= \left(\frac{Ac_2}{k+1} - \frac{Bc_1}{k}\right)\frac{(\log N)^k}{\log\log N}
\end{aligned}$$

ここで c_2 を $c_2^{\delta-1/2} \geq \dfrac{4}{\log 2}$ をみたし, かつ上の式の括弧内の定数が正になるようにとる. そして括弧内の定数を c_3 とおく. さらに

$$\frac{k-1}{2k} > t \geq 1-\delta$$

をみたすように t をとって,

$$\gamma = (N-1)(N^2-1)\dots(N^{[x^t]}-1)$$

を考える. γ は $N^{[x^t]}$ の大きさ以下の $[x^t]$ 個の数の積だから,

$$\begin{aligned}
\gamma \text{ の素因子の数} \leq \gamma \text{ の約数の数} &\sim x^t \log N^{x^t} \\
&= x^{2t}\log N = c_2^{2t}(\log N)^{2tk+1} \\
&< c_2^{2t}(\log N)^k \quad \because \frac{k-1}{2k} > t.
\end{aligned}$$

この数は, N を十分大きくとると, 上で得られた下限 $c_3\dfrac{(\log N)^k}{\log\log N}$ より小さい. よって

$$\exists \text{ 素数 } r \in [c_1(\log N)^k, c_2(\log N)^k]) \begin{cases} \gamma \not\equiv 0 \pmod r \\ r-1 \text{ の最大素因子は 補題 3 の条件をみたす.} \end{cases}$$

このとき $r-1$ の最大素因子 q は

$$\begin{aligned} q &\geq (c_2(\log N)^k)^\delta && \because \text{補題 3} \\ &\geq \frac{4c_2^{1/2}}{\log 2}(\log N)^{k\delta} && \because c_2^{\delta-1/2} \geq 4/\log 2 \\ &\geq \frac{4c_2^{1/2}}{\log 2}(\log N)^{k/2+1} && \because (4.1) \\ &= 4c_2^{1/2}(\log N)^{k/2}\log_2 N \\ &\geq 4\sqrt{r}\log_2 N && \because c_2(\log N)^k \geq r \end{aligned}$$

となり (P1) をみたす.

さらに

$$\begin{aligned} \frac{r-1}{q} &\leq \frac{c_2(\log N)^k}{(c_2(\log N)^k)^\delta} \\ &= (c_2(\log N)^k)^{1-\delta} \\ &\leq (c_2(\log N)^k)^t && \because 1-\delta \leq t \\ &= x^t \end{aligned}$$

より, $\gamma \not\equiv 0 \pmod r$ を考えると, (P2) も OK. □

この定理から r は $(\log N)^k$ の大きさで, s は q の大きさと同じ $(\log N)^{k/2+1}$ の大きさでとれることがわかる.

注意 6. (4.1) から, 補題 3 の δ が大きくとれれば, k が小さくとれて, 計算量の評価はよくなる. δ で現在もっとも良い評価は Fouvry [6] の $\delta = 0.6683$ である. 原論文ではこの結果を使って,

$$\delta = \frac{2}{3},\ t = \frac{1}{3},\ k = 6 \tag{4.2}$$

として上の証明を行なっている.

実際の計算では, (4) の二項係数を評価して r, s をとれば良い. どのくらいの大きさで r, s がとれるかを以下に例示しておく.

N	r	q	s
100003	1187	593	545
1000003	1619	809	793
10000019	2207	1103	1045
100000007	2879	1439	1386
1000000007	3623	1811	1785
10000000019	4547	2273	2187
100000000003	5387	2693	2650
1000000000039	6599	3299	3169
10000000000037	7523	3761	3676

5 計算量と実装例

5.1 計算量

命題 7. *AKS* のアルゴリズムの計算量は $O\left((\log N)^{\frac{5}{2}k+4}\right)$ である.

証明. 19行目からのループがこのアルゴリズムの計算量を決定する.

$(x-a)^N \bmod (x^r-1, N)$ の計算には $\mathbb{Z}/N\mathbb{Z}[x]$ の r 次以下の多項式のかけ算を $O(\log N)$ 回が必要.

$\mathbb{Z}/N\mathbb{Z}[x]$ の r 次以下の多項式のかけ算をするには, $\mathbb{Z}/N\mathbb{Z}$ でのかけ算が

FFT なし	FFT あり
$O(r^2)$	$O(r(\log N)^{\epsilon})$

回必要.

$\mathbb{Z}/N\mathbb{Z}$ の一回のかけ算に, 必要な基本的な演算の回数は

FFT なし	FFT あり
$O((\log N)^2)$	$O((\log N)^{1+\epsilon})$.

したがって問題のループには $O((\log N)^{k/2+1} \cdot r^2 \cdot \log N \cdot (\log N)^2)$ かかり, r が $O((\log N)^k)$ だから, あわせて $O\left((\log N)^{\frac{5}{2}k+4}\right)$ (FFT をつかうと $O\left((\log N)^{\frac{3}{2}k+3+\epsilon}\right)$) となる. □

パラメータを (4.2) のようにとると, 計算量は $O\left((\log N)^{19}\right)$ である (FFT をつかうと $O((\log N)^{12+\epsilon})$). 論文 [1] では Sophie Germain 素数の分布に関するある予想を仮定すると, 計算量は $O((\log N)^{6+\epsilon})$ になることが示されている.

FFT (Fast Fourier Transform) については [5] を見よ.

5.2 実装例

[4] をみるといろいろな実装が発表されている. ここでは Allan K. Steel の MAGMA への実装をもとに Bernstein の改良を含めた形のプログラムで実行例を見る. ただしこの実装では r を求める部分で組み込み関数である `NextPrime` を使っているので厳密には決定的でない. しかしながら, このアルゴリズムがどのように動作するかを見るには十分である. 以下の実行時間は Mobile Pentium III 866MHz でのものである.

```
> AKS(100003);
r= 1187 s= 545
Selection of r and s: 6.57
Final loop: 93.65
true
> AKS(1000003);
r= 1619 s= 793
Selection of r and s: 12.21
Final loop: 214.429
```

```
true
> AKS(10000019);
r= 2207 s= 1045
Selection of r and s: 23.37
Final loop: 1044.701
true
```

これでわかるようにAKSのアルゴリズムは実用的なものではなく，理論的な面にその重要性がある．

参考文献

[1] M. Agrawal, N. Kayal, and N. Saxena, *PRIMES is in P*, Preprint. (2002). http://www.cse.iitk.ac.in/news/primality.html.

[2] R. C. Baker and G. Harman, *The Brun-Titchmarsh theorem on average*, Analytic number theory, Vol. 1 (Allerton Park, IL, 1995), Progr. Math., vol. 138, Birkhäuser Boston, Boston, MA, 1996, pp. 39–103.

[3] D. J. Bernstein, *An exposition of the Agrawal-Kayal-Saxena primality-proving theorem*, Preprint. (2002). http://cr.yp.to/papers/aks.ps.

[4] P. Cramody, *Links to things relevant to the AKS algorithms.* http://fatphil.asdf.org/maths/AKS.

[5] R. Crandall and C. Pomerance, *Prime numbers — A computational perspective*, Springer-Verlag, New York, 2001.

[6] É. Fouvry, *Théorème de Brun-Titchmarsh: application au théorème de Fermat*, Invent. Math. **79** (1985), no. 2, 383–407.

[7] M. Goldfeld, *On the number of primes p for which p + a has a large prime factor*, Mathematika **16** (1969), 23–27.

[8] S. Goldwasser and J. Kilian, *Primality testing using elliptic curves*, J. ACM **46** (1999), no. 4, 450–472.

[9] G. Jaeschke, *On strong pseudoprimes to several bases*, Math. Comp. **61** (1993), no. 204, 915–926.

[10] A. Klappenecker, *The AKS primality test — results from analytic number theory.* http://faculty.cs.tamu.edu/klappi/629/analytic.pdf.

[11] N. Koblitz, *Algebraic aspects of cryptography*, Algorithms and Computation in Mathematics, vol. 3, Springer-Verlag, Berlin, 1998, With an appendix by Alfred J. Menezes, Yi-Hong Wu and Robert J. Zuccherato.

[12] P. Luschny, *AKS, a deterministic primary test*, Maple V program. http://www.luschny.de/math/primes/aks.txt.

[13] G. L. Miller, *Riemann's hypothesis and tests for primality*, J. Comput. System Sci. **13** (1976), no. 3, 300–317, Working papers presented at the ACM-SIGACT Symposium on the Theory of Computing (Albuquerque, N.M., 1975).

[14] M. O. Rabin, *Probabilistic algorithm for testing primality*, J. Number Theory **12** (1980), no. 1, 128–138.

[15] 岡本龍明, 太田和夫（編）, 暗号・数論・ゼロ知識証明, 共立出版, 1995.

[16] A. Stiglic, *The PRIMES is in P little FAQ*.
`http://crypto.cs.mcgill.ca/~stiglic/PRIMES_P_FAQ.html`.

[17] 内山成憲, *PRIMES is in P —- the aks primality testing*, JANT8.
`http://ntw.e-one.uec.ac.jp/jant/program08.html`.

(講演 2002 年 12 月 2 日)

e-mail: kida@sugaku.e-one.uec.ac.jp